

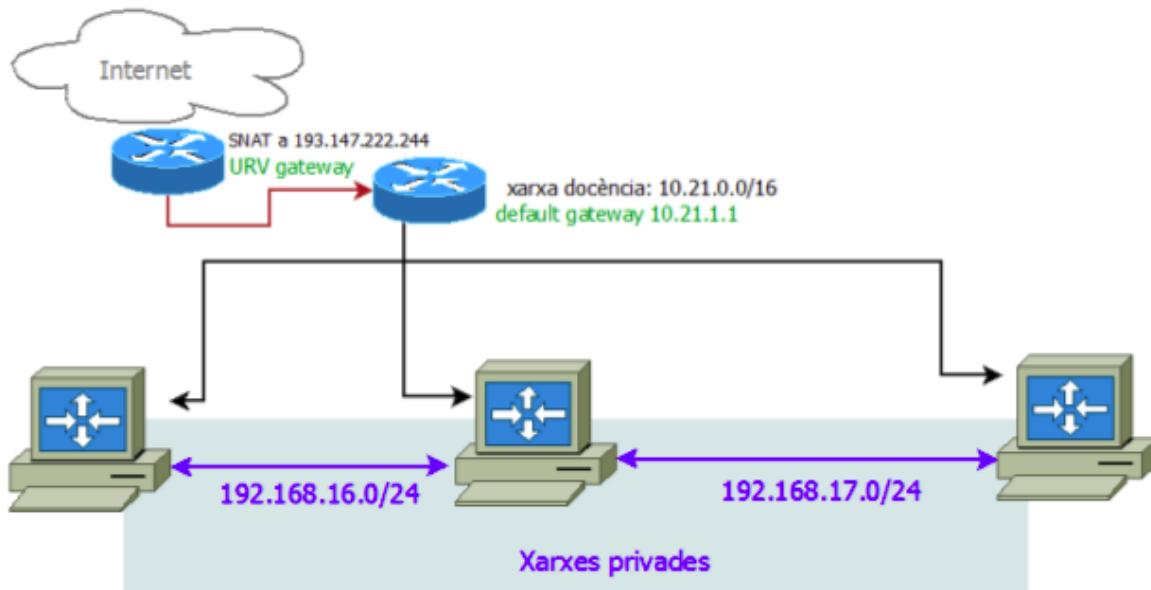
# Lab 1: Repàs i eines per Xarxes

## Objectiu:

L'objectiu d'aquesta primera pràctica és el de crear dues xarxes privades entre 3 màquines, en el nostre cas 3 màquines virtuals. Una actuarà de router, sent el punt de connexió entre les dues xarxes i les altres faran una de servidor i l'altra de client.

Tasques:

- Configurar dues xarxes privades amb adreces estàtiques sobre les ethernets lliures dels 3 PCs, tal com s'indica a l'esquema.
- Utilització de diferents eines de xarxa: ifconfig, eth-tool, mii-tool, ping, traceroute, wireshark.



Per automatitzar la configuració en les tres màquines hem implementat un únic script, de forma que introduint com a argument la màquina (router, client o server) i les interfícies de xarxa (sinó se'n seleccionaran per defecte) s'escriu la configuració desitjada en el fitxer /etc/network/interfaces, es configuren les iptables (en el router) i es ressetegen tots els serveis i interfícies de xarxa.

En aquesta primera fase, a més de crear les dues xarxes (una que connecta el router amb el servidor i l'altra que connecta el router amb el client) i permetre el forwarding de paquets per comunicar servidor i client a través del router, hem configurat la NAT en el router per permetre la connexió a internet a través seu.

## Script de configuració:

Com ja hem comentat anteriorment, l'script de configuració ha consistit en un únic fitxer on especificuem la màquina que estem configurant en aquell moment. L'estructura bàsica és la següent, primer hi ha definides les funcions de configuració per cada màquina i una funció de mostra ajuda. Al executar l'script el que primer es fa és mirar si s'ha seleccionat l'opció

d'ajuda, en cas afirmatiu, aquesta es mostra i acaba l'execució. Si no és selecciona l'opció d'ajuda es mirarà quina de les màquines és vol configurar, si la selecció no és correcta mostra un codi d'error i acaba l'execució.

### Estructura de l'script

```
function display_help(){  
}  
function router_config(){  
}  
function client_config(){  
}  
function server_config(){  
}  
while getopts :h option  
do  
    case "$option" in  
        h)  
            display_help  
            exit 1  
            ::  
        *)  
            echo "ERROR: Invalid parameters" >&2  
            display_help  
            exit 1  
            ::  
    esac  
done  
  
case "$1" in  
    router)  
        router_config  
        exit 0  
        ::  
    client)  
        client_config  
        exit 0  
        ::  
    server)  
        server_config  
        exit 0  
        ::  
    *)  
        echo -e "ERROR: machine indication needed:\n\trouter\n\tclient\n\tserver"  
        exit 1  
esac
```

## Configuració Router:

Si el primer argument d'entrada és "router" s'executarà el següent codi. Primer es comprova si s'han introduït 3 arguments addicionals (correspondents a 3 interfícies de xarxa), si no s'han introduït es seleccionaran d'entre les existents.

Seguidament s'escriu en el fitxer /etc/network/interfaces la configuració que es vol assignar a cada interfície de xarxa, es força l'apagat de les interfícies de xarxa, s'activa el forwarding de paquets (escrivint un 1 en el fitxer /proc/sys/net/ipv4/ip\_forward) i es tornen a activar les interfícies de xarxa.

Finalment, configurem les iptables per permetre la connexió a Internet de les dues xarxes a les quals està connectat el router (servidor i client podran connectar-se a Internet a través del router).

### Codi Router

```
function router_config(){
if [ $# -lt 4 ]
then
    echo "Usem la configuració per defecte"
    #Llistem les interfícies de xarxa
    interfaces=$(ip address show | egrep -e "[0-9]" | cut -d ':' -f2 | tr -d ' ' | sed '/lo/d')
    i1=$(echo $interfaces | cut -d ' ' -f1)
    i2=$(echo $interfaces | cut -d ' ' -f2)
    i3=$(echo $interfaces | cut -d ' ' -f3)
else
    i1=$2
    i2=$3
    i3=$4
fi

config="auto lo\niface lo inet loopback
\n
\nauto $i1\niface $i1 inet dhcp
\n
\nallow-hotplug $i2\niface $i2 inet static\n\taddress 192.168.16.1\n\tinetmask
255.255.255.0\n\tnetwork 192.168.16.0\n\tbroadcast 192.168.16.255
\n
\nallow-hotplug $i3\niface $i3 inet static\n\taddress 192.168.17.1\n\tinetmask
255.255.255.0\n\tnetwork 192.168.17.0\n\tbroadcast 192.168.17.255"

#Apaguem les interfícies de xarxa
ifdown $i1 --force
ifdown $i2 --force
ifdown $i3 --force

#escribim canvis en el fitxer /etc/network/interfaces
echo -e $config > /etc/network/interfaces

#Activem el forwarding
echo 1 >/proc/sys/net/ipv4/ip_forward
#reiniciem les interfícies de xarxa
ifup $i1
ifup $i2
ifup $i3
```

```

#obtenim la ip amb conexio a internet del router
$addres=$(hostname -I | cut -d ' ' -f1)

#permetem el tràfic cap a l'exterior
iptables -I INPUT -j ACCEPT
iptables -t nat -A POSTROUTING -s 192.168.16.0/24 -j SNAT --to $addres
iptables -t nat -A POSTROUTING -s 192.168.17.0/24 -j SNAT --to $addres
}

```

Un cop executat l'Script de configuració, llistant les interfícies de xarxa s'observen com estan connectats i configurats als ports ethernet del Router les dues xarxes.

```

milax@casa:~/GSX_Labs/Prac1Xarxes$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
        inet 127.0.0.1/8 scope host lo
            valid_lft forever preferred_lft forever
        inet6 ::1/128 scope host
            valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UNKNOWN group default
qlen 1000
    link/ether 08:00:27:c1:34 brd ff:ff:ff:ff:ff:ff
        inet 10.0.2.15/24 brd 10.0.2.255 scope global eth0
            valid_lft forever preferred_lft forever
        inet6 fe80::a00:27ff:fe41:c134/64 scope link
            valid_lft forever preferred_lft forever
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UNKNOWN group default
qlen 1000
    link/ether 08:00:27:c5:da:83 brd ff:ff:ff:ff:ff:ff
        inet 172.17.2.1/24 brd 172.17.2.255 scope global eth1
            valid_lft forever preferred_lft forever
        inet6 fe80::a00:27ff:fec5:da83/64 scope link
            valid_lft forever preferred_lft forever
4: eth2: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UNKNOWN group default
qlen 1000
    link/ether 08:00:27:d4:e2:3d brd ff:ff:ff:ff:ff:ff
        inet 192.168.8.1/23 brd 192.168.9.255 scope global eth2
            valid_lft forever preferred_lft forever
        inet6 fe80::a00:27ff:fed4:e23d/64 scope link
            valid_lft forever preferred_lft forever

```

## Configuració Server-Client:

En canvi, si el primer argument d'entrada és “server” o “client” s'executarà un dels codis següents (només varien en la xarxa a la qual es connecten, 192.168.16.0 i 192.168.17.0 respectivament). Primer es comprova si s'ha introduït un argument addicional referent a la interfície de xarxa a configurar. Si no s'ha introduït es seleccionarà d'entre les existents. Seguidament s'escriu en el fitxer /etc/network/interfaces la configuració adient, es força l'apagat de la interfície de xarxa i es torna a activar.

Finalment, s'afegeixen les rutes necessàries per poder connectar-te des d'una xarxa a l'altra a través del router i per poder-te connectar a Internet.

### Codi Client i Servidor

```

function client_config(){
if [ $# -lt 2 ]
then
    echo "Usem la configuració per defecte"

#Llistem les interfícies de xarxa

```

```

interfaces=$(ip address show | egrep -e "^[0-9]" | cut -d ':' -f2 | tr -d ' ' | sed '/lo/d')
i1=$(echo $interfaces | cut -d ' ' -f1)

else
    i1=$2
fi

#revisar la configuració de l'adreça
config="auto lo\niface lo inet loopback
        \n
        \nallow-hotplug $i1\niface $i1 inet static\n\taddress 192.168.17.2\n\tinetmask
255.255.255.0\n\tbroadcast 192.168.17.255"

ifdown $i1 --force

#escribim canvis en el fitxer /etc/network/interfaces
echo -e $config > /etc/network/interfaces

#Afegim les rutes que calguin
echo "up ip route add 192.168.16.0/24 via 192.168.17.1 dev $i1" >> /etc/network/interfaces
echo "up ip route add default via 192.168.17.1" >> /etc/network/interfaces

ifup $i1
}

function server_config(){
if [ $# -lt 2 ]
then
    echo "Usem la configuració per defecte"
    echo "ERROR, you have to introduce an argument" >&2

    #Llistem les interfícies de xarxa
    interfaces=$(ip address show | egrep -e "^[0-9]" | cut -d ':' -f2 | tr -d ' ' | sed '/lo/d')
    i1=$(echo $interfaces | cut -d ' ' -f1)

else
    i1=$2
fi

#revisar la configuració de l'adreça
config="auto lo\niface lo inet loopback
        \n
        \nallow-hotplug $i1\niface $i1 inet static\n\taddress 192.168.16.2\n\tinetmask
255.255.255.0\n\tbroadcast 192.168.16.255"

ifdown $i1 --force

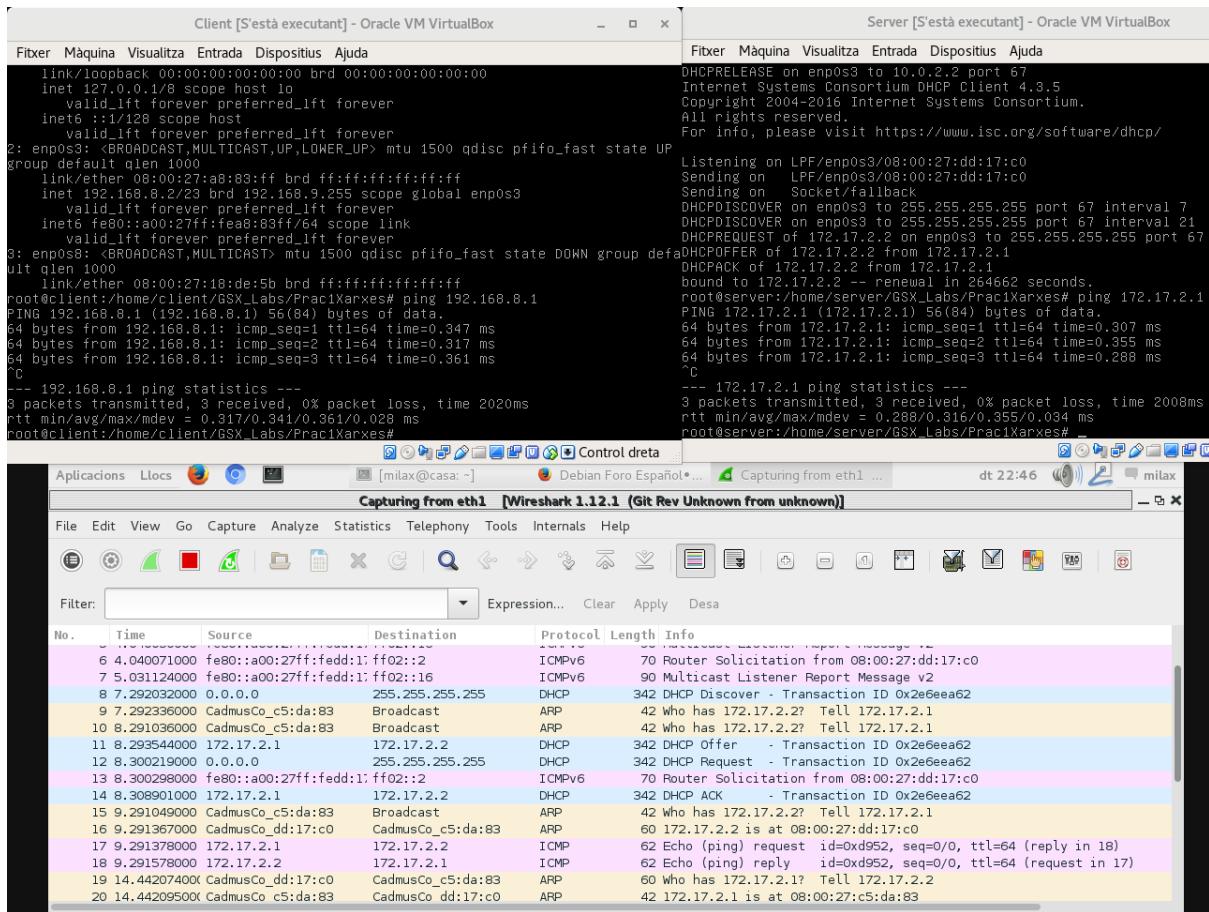
#escribim canvis en el fitxer /etc/network/interfaces
echo -e $config > /etc/network/interfaces

#Afegim les rutes que calguin
echo "up ip route add 192.168.17.0/24 via 192.168.16.1 dev $i1" >> /etc/network/interfaces
echo "up ip route add default via 192.168.16.1" >> /etc/network/interfaces

ifup $i1
}

```

Es realitzen pings entre les maquines Server i Client i es corrobora el trànsit de paquets entre les dos xarxes mitjançant el Wireshark a la màquina central (Router), com es pot constatar en la següent captura.



## PREGUNTES

**Proveu a fer un ping des de cada una de les màquines extremes a la màquina central (utilitzant la IP que aquesta té en la mateixa xarxa on ens trobem).**

Funciona correctament tant el ping de les màquines extremes a la central i viceversa com el ping d'una màquina extrema a l'altra i viceversa.

A més, també es permet fer ping des de les màquines extremes a google (8.8.8.8) sent el router (màquina central) l'únic que té comunicació a Internet.

**Engegueu el wireshark en la màquina central i repetiu el ping que heu fet. Quin és el protocol que utilitzen els paquets de tipus ping?**

El protocol que usen els paquets de tipus ping és el ICMP (Internet Message Control Protocol)

**Des del PC de l'esquerra proveu a fer: ping 192.168.17.1 Funciona? Perquè?**

Sí, ja que hem configurat una ruta de forma que qualsevol paquet que vagi a l'altra xarxa ho faci a través del router, i aquesta IP a la qual fem ping és la del router en l'altra xarxa.

**PISTA: Proveu a fer traceroute en comptes de ping . Per on passa el paquet?**

S'observa com passa pel router.

```
root@debian-gnu-linux-8:/home/parallels/Downloads/GSX_Labs-master/Prac1Xarxes# traceroute 192.168.17.2
traceroute to 192.168.17.2 (192.168.17.2), 30 hops max, 60 byte packets
 1  192.168.16.1 (192.168.16.1)  0.304 ms  0.286 ms  0.277 ms
 2  192.168.17.2 (192.168.17.2)  0.895 ms  0.892 ms  1.045 ms
```

### **Quants Hosts podriem tenir com a màxim en cada una d'aquestes xarxes?**

Tenim 8 bits dedicats al host, el qual dóna lloc a  $2^8 = 256$  hosts diferents. Degut a que hi ha una adreça broadcast reservada i una altra que no apunta a cap altre host, sinó que és la adreça efectiva de la xarxa on ens trobem, això dóna un total de  $256 - 2 = 254$  hosts totals.

## **Supernetting**

El **supernetting** és un procediment que aprofita els principis de CIDR per a direccionar cap a un conjunt de xarxes IP utilitzant una sola ruta. Aquesta ruta obtinguda, s'anomena supernet.

**Ara mateix tenim una configuració amb dues xarxes locals, amb IPs privades de rangs diferents. Sense modificar aquests rangs, feu que tots 2 rangs formin part de la mateixa xarxa. Com a idea podeu veure que els 2 bytes últims de les IPs de la Universitat (10.21.xx.xx) formen la mateixa xarxa. Perquè és això?**

El supernetting es procediment que ens permet agregar múltiples xarxes IPs de la mateixa classe utilitzant una sola ruta. Utilitzant supernetting, l'adreça 192.168.2.0/24 i l'adreça adjacent 192.168.3.0/24 poden ser combinades en la xarxa "lògica" 192.168.2.0/23, anomenada superxarxa (supernet). Els canvi de 24 a 23 bits a 1 de la màscara de xarxa indiquen que ara es dedica un bit més als hosts. D'aquesta manera, es creen dues subxarxes. Això permet separar el tràfic, oferint una major seguretat degut a la possibilitat d'aïllar els problemes en una sola subxarxa.

També possibilita que l'administrador controli la mida que vol que tingui cada subxarxa. Això és interessant, per exemple, si hem d'administrar la xarxa d'un edifici sencer: Segons la quantitat d'ordinadors que tinguem a cada planta podem crear una mida de xarxa dependent del nombre de bits que dediquem als hosts.

Degut a la separació virtual de les diferents subxarxes també permet reduir la grandària de les taules d'encaminament. Es dediquen més bits a la definició dels usuaris, pel que podem tenir més maquines connectades a una mateixa xarxa.

**Com podeu modificar de manera senzilla la vostra configuració per tal que també les 2 xarxes que hem creat siguin una única xarxa de tipus 192.168.xx.xx ?**

La configuració addicional necessària implica una màscara de supernet a l'hora de configurar les adreces IP de les dos xarxes. Aquesta mascara ha de proporcionar un rang d'adreses en el que com a mínim es puguin configurar dos xarxes. Per tant, es considera la mascara de /23 bits, que permet adreces per a un màxim de dues xarxes: **255.255.254.0**

Es modifica la mascara en l'especificació *tinetmask*, així com la *tnetwork* per incloure-ho dins la mateixa xarxa de la configuració ip dels tres components de la xarxa, a les funcions *server\_config*, *client\_config*, *router\_config* del *initial\_config.sh*:

### **Codi Router Client i Servidor**

```
function client_config(){
....
```

```

\nallow-hotplug $i1\niface $i1 inet static\n\taddress 192.168.16.2\n\tinetmask
255.255.254.0\n\tnetwork 192.168.16.0\n\tbroadcast 192.168.17.255"
....}
}

function server_config(){
...
\nallow-hotplug $i1\niface $i1 inet static\n\taddress 192.168.16.2\n\tinetmask
255.255.254.0\n\tnetwork 192.168.16.0\n\tbroadcast 192.168.16.255"
....}
}

function router_config(){
...
\nallow-hotplug $i2\niface $i2 inet static\n\taddress 192.168.16.1\n\tinetmask
255.255.254.0\n\tnetwork 192.168.16.0\n\tbroadcast 192.168.16.255

\nallow-hotplug $i3\niface $i3 inet static\n\taddress 192.168.17.1\n\tinetmask
255.255.254.0\n\tnetwork 192.168.17.0\n\tbroadcast 192.168.17.255"
....}
}

```

En la captura següent s'observa com s'ha aplicat la màscara de supernetting correctament, mostrant un /23 adjacent a l'adreça IP.

```

Debian GNU Linux 8
Activities Terminal Sat 16:12
parallels@debian-gnu+linux-8: ~/Downloads/GSX_Labs-master/PraciXarxes

File Edit View Search Terminal Help
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
        inet 127.0.0.1/128 scope host
            valid_lft forever preferred_lft forever
2: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 00:1c:42:0c:65:bb brd ff:ff:ff:ff:ff:ff
    valid_lft forever preferred_lft forever
    inet 192.168.16.1/23 brd 192.168.16.255 scope global dynamic eth1
        valid_lft 172200sec preferred_lft 3152sec
    inet 192.168.16.1/23 brd 192.168.16.255 scope global secondary eth1
        valid_lft forever preferred_lft forever
    inet 192.168.17.1/23 brd 192.168.17.255 scope global secondary eth1
        valid_lft forever preferred_lft forever
    inet 192.168.17.1/23 brd 192.168.17.255 scope global temporary dynamic
        valid_lft 172200sec preferred_lft 3152sec
    inet6 fe80::1c42:65ff:fe0c:65bb/64 scope global mngtmpaddr noprefroute dynamic
        valid_lft 2591999sec preferred_lft 694599sec
    inet6 fe80::2c26:144f:1c42:ff:fe0c:65bb/64 scope global mngtmpaddr noprefroute dynamic
        valid_lft 2591999sec preferred_lft 694599sec
    inet6 fe80::2c26:144f:1c42:ff:fe0c:65bb/64 scope link
        valid_lft forever preferred_lft forever
root@ubuntu:/home/parallels/Desktop/GSX_Labs-master/PraciXarxes# ifconfig
Open a configuration file
ERROR, you have to introduce an argumnet
root@ubuntu:/home/parallels/Desktop/GSX_Labs-master/PraciXarxes# ./initial_config_supernetting.sh server
Usen la configuració per defecte
ERROR, you have to introduce an argumnet
root@ubuntu:/home/parallels/Desktop/GSX_Labs-master/PraciXarxes# ./initial_config_supernetting.sh client
ERROR, you have to introduce an argumnet
root@ubuntu:/home/parallels/Desktop/GSX_Labs-master/PraciXarxes# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 1500 qdisc noqueue state UNKNOWN group default qlen 1
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
        inet 127.0.0.1/128 scope host
            valid_lft forever preferred_lft forever
2: enp0s5: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 00:1c:42:fa:2f:ff brd ff:ff:ff:ff:ff:ff
    inet 10.37.129.3/24 brd 10.37.129.255 scope global dynamic enp0s5
        valid_lft 1797sec preferred_lft 1797sec
    inet 192.168.16.2/23 brd 192.168.16.255 scope global enp0s5
        valid_lft forever preferred_lft forever
root@ubuntu:/home/parallels/Desktop/GSX_Labs-master/PraciXarxes# ifconfig

```

Funciona ara un ping des del PC de l'esquerra fins a la 192.168.17.1? Per què?

**Feu un traceroute. Per on passa el paquet?**

En la captura següent s'observa com amb la comanda ping s'estableix connexió amb el router (adreça IP acabada amb .1). Funciona ja que, a part de tenir configurada la xarxa correctament amb el router, d'adreça indicada esta dins el rang que proporciona la mascara de la supernet.

```
buntu:/home/parallels/Desktop/GSX_Labs-master/Prac1X En (0) 7:23 AM ↵
rtt min/avg/max/mdev = 0.494/0.679/0.891/0.131 ms
root@ubuntu:/home/parallels/Desktop/GSX_Labs-master/Prac1Xarxes# ping 192.168.17.1
PING 192.168.17.1 (192.168.17.1) 56(84) bytes of data.
64 bytes from 192.168.17.1: icmp_seq=1 ttl=64 time=0.257 ms
64 bytes from 192.168.17.1: icmp_seq=2 ttl=64 time=0.335 ms
64 bytes from 192.168.17.1: icmp_seq=3 ttl=64 time=0.219 ms
64 bytes from 192.168.17.1: icmp_seq=4 ttl=64 time=0.258 ms
64 bytes from 192.168.17.1: icmp_seq=5 ttl=64 time=0.395 ms
^C
--- 192.168.17.1 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 3997ms
rtt min/avg/max/mdev = 0.219/0.298/0.395/0.067 ms
root@ubuntu:/home/parallels/Desktop/GSX_Labs-master/Prac1Xarxes# ping 192.168.17.1
PING 192.168.17.1 (192.168.17.1) 56(84) bytes of data.
64 bytes from 192.168.17.1: icmp_seq=1 ttl=64 time=0.412 ms
64 bytes from 192.168.17.1: icmp_seq=2 ttl=64 time=0.316 ms
64 bytes from 192.168.17.1: icmp_seq=3 ttl=64 time=0.311 ms
64 bytes from 192.168.17.1: icmp_seq=4 ttl=64 time=0.302 ms
64 bytes from 192.168.17.1: icmp_seq=5 ttl=64 time=0.375 ms
64 bytes from 192.168.17.1: icmp_seq=6 ttl=64 time=0.428 ms
^C
--- 192.168.17.1 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 4999ms
rtt min/avg/max/mdev = 0.302/0.357/0.428/0.052 ms
root@ubuntu:/home/parallels/Desktop/GSX_Labs-master/Prac1Xarxes#
```

**Feu un traceroute. Per on passa el paquet?**

S'observa com passa pel router.

```
root@debian-gnu-linux-8:/home/parallels/Downloads/GSX_Labs-master/Prac1Xarxes# traceroute 192.168.17.2
traceroute to 192.168.17.2 (192.168.17.2), 30 hops max, 60 byte packets
 1  192.168.16.1 (192.168.16.1)  0.304 ms  0.286 ms  0.277 ms
 2  192.168.17.2 (192.168.17.2)  0.895 ms  0.892 ms  1.045 ms
```

Tenint en compte els rangs d'adreses IPs que utilitzàvem en les 2 xarxes privades, esteu segurs que la modificació que heu fet és la més òptima?

Si és la més adient ja que permet un adreçament per a màxim 2 xarxes, sent així òptim per una xarxa de 2 xarxes privades que és el que tenim actualment.

**Finalment proveu a fer ping des del PC de l'esquerra al PC de la dreta. Funciona? Perquè?**

També funciona en aquest cas, ja que el PC configurat com a server (IP 192.168.17.2) està configurat dins la supernet, amb la adreça IP dins el rang que permet la mascara. Com es constata en la captura a continuació, el ping funciona en ambdues direccions:

En aquesta nova xarxa, des de quina IP fins a quina IP tenim disponible per als seus Hosts? Quants Hosts podem arribar a tenir, doncs?

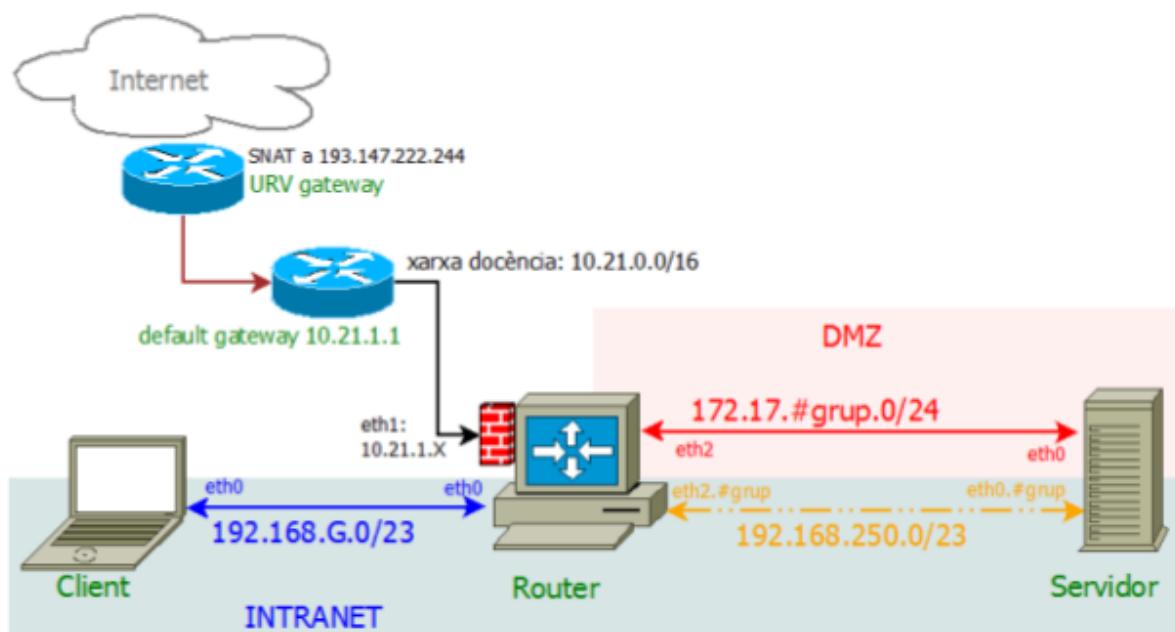
El nombre hosts que proporciona el rang d'adreces permès per la mascara de la supernet es de 510, ja que tenim disponibles adreçar dos subxarxes amb 255 hosts cadascuna (en total 510), des de 192.168.16.1 fins a 192.168.17.254, tenint el doble de *Hosts* que es poden connectar a la mateixa xarxa, ja que s'ha dedicat un bit més a la definició d'aquests.

# Lab 2: Configuració de xarxa i DHCP bàsic

## Objectiu:

Dissenyar un esquema d'adreçament privat per a la nostra DMZ i intranet.

## Tasques a realitzar:



- Configurarem la màquina central (àlies router) amb IPs estàtiques i amb el servei DHCP. Aquesta màquina serà el nostre servidor de DHCP.
- Configurarem les altres 2 màquines per a que obtinguin la IP per DHCP.
- Prepararem un script per a automatitzar-ho de cara a les sessions posteriors.

El que primer hem de fer, és la màquina que actuarà de router instal·lar el paquet `isc-dhcpd-server` (això només cal fer-ho el primer cop).

Hem modificat el fitxer de configuració `/etc/dhcp/dhcpd.conf` per a que quedi de la següent forma:

### **dhcpd.conf**

```
# opciones comunes a todas las subxarxes
default-lease-time 14400; max-lease-time 86400;
authoritative;

# Configuracio DMZ:
subnet 172.17.2.0 netmask 255.255.255.0 {
range 172.17.2.3 172.17.2.254;
option subnet-mask 255.255.255.0;
option broadcast-address 172.17.2.255;
option routers 172.17.2.1;
option domain-name "DMZ_2.gs";
option domain-name-servers 172.17.2.1;
max-lease-time 604800;
default-lease-time 604800;
# Configuracio Statica Server
host server{
```

```

hardware ethernet 08:00:27:dd:17:c0;
fixed-address 172.17.2.2;
}

}

# Configuracio INTRANET:
subnet 192.168.8.0 netmask 255.255.254.0 {
range 192.168.8.3 192.168.9.254;
option subnet-mask 255.255.254.0;
option broadcast-address 192.168.9.255;
option routers 192.168.8.1;
option domain-name "INTRANET";
option domain-name-servers 192.168.8.1;
max-lease-time 604800;
default-lease-time 604800;
}

```

S'observa com per una banda configurem una de les xarxes (DMZ) que treballa en la subxarxa 172.17.2.0, amb IPs permeses de la 3 a la 254, ja que la .1 està assignada al router, la .2 s'assigna de form estàtica al servidor i la .255 correspon al broadcast.

D'altra banda tenim la xarxa INTRANET, on s'ha fet supernetting (s'assignen 9 bits per descriure els hosts).

A més, afegim les interfícies a /etc/default/isc-dhcp-server.

Hem reaprofitat l'script de la fase anterior (només mostrem la part modificada):

### Codi router

```

if [ $# -lt 4 ]
then
    echo "Usem la configuració per defecte"

    #Llistem les interfícies de xarxa
    interfaces=$(ip address show | egrep -e "^[0-9]" | cut -d ':' -f2 | tr -d ' ' | sed '/lo/d')
    i1=$(echo $interfaces | cut -d ' ' -f1)
    i2=$(echo $interfaces | cut -d ' ' -f2)
    i3=$(echo $interfaces | cut -d ' ' -f3)

else
    i1=$2
    i2=$3
    i3=$4
fi

config="auto lo\niface lo inet loopback
        \n
        \nauto $i1\niface $i1 inet dhcp
        \n
        \nallow-hotplug $i2\niface $i2 inet static\n\taddress 192.168.16.1\n\tinetmask
255.255.255.0\n\tnetwork 192.168.16.0\n\tbroadcast 192.168.16.255
        \n
        \nallow-hotplug $i3\niface $i3 inet static\n\taddress 192.168.17.1\n\tinetmask
255.255.255.0\n\tnetwork 192.168.17.0\n\tbroadcast 192.168.17.255"

```

```
cp -p dhcpcd.conf /etc/dhcp/dhcpcd.conf
```

```

#Afegim les interfícies en /etc/default/isc-dhcp-server
sed -i "s/INTERFACES=\".*\"/INTERFACES=\"$i2 $i3\"/g" "/etc/default/isc-dhcp-server"

#Apaguem les interfícies de xarxa
ifdown $i1 --force
ifdown $i2 --force
ifdown $i3 --force

#escribim canvis en el fitxer /etc/network/interfaces
echo -e $config > /etc/network/interfaces

#Activem el forwarding
echo 1 >/proc/sys/net/ipv4/ip_forward

#reiniciem les interfícies de xarxa
ifup $i1
ifup $i2
ifup $i3

#obtenim la ip amb conexió a internet del router
addres=$(hostname -I | cut -d ' ' -f1)

#permetem el tràfic cap a l'exterior
iptables -I INPUT -j ACCEPT
iptables -t nat -A POSTROUTING -s 192.168.8.0/23 -j SNAT --to $addres
iptables -t nat -A POSTROUTING -s 172.17.2.0/24 -j SNAT --to $addres

```

En el servidor i client, només hem hagut de modificar el fitxer interfaces, on enlloc d'assignar la IP de forma estàtica, ho fem a través de DHCP, a més com les rutes ja s'assignen per defecte, no les afegim.

#### Codi client/servidor

```

if [ $# -lt 2 ]
then
    echo "Usem la configuració per defecte"

    #Llistem les interfícies de xarxa
    interfaces=$(ip address show | egrep -e "^[0-9]" | cut -d ':' -f2 | tr -d ' ' | sed '/lo/d')
    i1=$(echo $interfaces | cut -d ' ' -f1)

else
    i1=$2
fi

config="auto lo\niface lo inet loopback
        \n
        \nallow-hotplug $i1\niface $i1 inet dhcp"

ifdown $i1 --force

#escribim canvis en el fitxer /etc/network/interfaces
echo -e $config > /etc/network/interfaces

```

```
#Afegim les rutes que calguin
echo "up ip route add 172.17.2.0/24 via 192.168.8.1 dev $i1" >> /etc/network/interfaces
echo "up ip route add default via 192.168.8.1" >> /etc/network/interfaces

ifup $i1
```

### PREGUNTES INTERESSANTS:

- Quants ordinadors podeu arribar a tenir connectats a la vegada en aquesta xarxa?  
En la nova xarxa entre client i router, en fer supernetting, podem connectar  $2^9 - 2 = 510$  màquines.
- Podríem ampliar més la xarxa sense col·lisionar amb els espais d'adreces dels altres grups? En cas que la resposta sigui afirmativa, quants bits més podríem agafar?  
Sí, amb la configuració actual (subnet /23) es poden tenir 510 hosts per xarxa i 128 subnets. D'aquesta forma, dedicant més bits a la configuració dels hosts (aplicant una màscara /22 o /21) podríem connectar moltes més màquines (1022 i 2046 respectivament), per contra, disminuirien el nombre de grups possibles, a més de que haurem de multiplicar per  $2^i$  el numero de grup per indicar la subxarxa (G) on i = el nombre de bits que dediquem al supernetting.

## VLAN

El servidor també donarà algun servei a la intranet. Per això usarem una VLAN (enllaç groc), i crearem una ethernet virtual a cada costat, separant “virtualment” el tràfic del de la DMZ. L'ID de la VLAN serà el vostre número de grup + 100 (si el grup fos el 7 llavors la ethernet virtual seria la eth2.107).

#### PER NOTA:

Instal·leu el paquet VLAN tant al servidor com al router i configureu-hi la xarxa de color groc de l'esquema.

Trieu sobre paper una IP per al router i una IP per al servidor (enllaç blau discontinu).

### Codi client/servidor

```
if [ $# -lt 2 ]
then
    echo "Usem la configuració per defecte"

    #Llistem les interfícies de xarxa
    interfaces=$(ip addr show | egrep -e "^[0-9]" | cut -d ':' -f2 | tr -d ' ' | sed '/lo/d')
    i1=$(echo $interfaces | cut -d ' ' -f1)

else
    i1=$2
fi

config="auto lo\niface lo inet loopback
        \n
```

```
\nallow-hotplug $i1\niface $i1 inet dhcp"\n\nifdown $i1 --force\n\n#escribim canvis en el fitxer /etc/network/interfaces\necho -e $config > /etc/network/interfaces\n\n#Afegim les rutes que calguin\necho "up ip route add 192.168.16.0/24 via 192.168.17.1 dev $i1" >> /etc/network/interfaces\necho "up ip route add default via 192.168.17.1" >> /etc/network/interfaces\n\nifup $i1
```

# Servidor DNS

## Objectiu:

- Ampliar el disseny de les xarxes de la pràctica anterior amb un servidor DNS.

- Configurar un servidor DNS primari responsable de varíes zones.

El primer que hem de fer en el servidor, és instal·lar el paquet bind9, que serà el nostre servidor DNS (això només és necessari el primer cop).

Modifiquem els fitxers de configuració /etc/bind/named.conf.local i /etc/bind/named.conf.options. En el primer dues zones, una per la INTRANET i una altra per la DMZ, a més de les 3 zones inverses corresponents (2 per la zona d'INTRANET ja que hem fet supernetting).

En l'altre fitxer s'especificarà el forwarding dels queries desconeguts cap al servidor DNS del ISP, bàsicament el que hem fet és copiar les IPs que hi ha en el fitxer /etc/resolv.conf del router a l'apartat forwarders.

## named.conf.local

```
#Options {  
#   directory "/etc/namedb";  
#   allow-query-cache { none; };  
#   allow-query { any; };  
#   recursion no;           #yes perquè funcionin els externs  
#};  
  
zone "grup2.gsx" {  
    type master;  
    file "DMZ_2.gsx.db";  
};  
  
zone "interna" {  
    type master;  
    file "INTRANET.db";  
};  
  
zone 2.17.172.in-addr.arpa {  
    type master;  
    file "db.172";  
};  
  
zone 8.168.192.in-addr.arpa{  
    type master;  
    file "db.192.168.8";  
};  
  
zone 9.168.192.in-addr.arpa{  
    type master;  
    file "db.192.168.9";  
};
```

Ara hem de crear un fitxer per cada zona definida en el named.conf.local, aquests fitxers es localitzaran en la carpeta /var/cache/bind/

## **DMZ\_2.gsx.db**

```
$TTL 604800
@ IN SOA ns.grup2.gsx. root.ns.grup2.gsx. (
    1 ; Serial
    604800 ; Refresh
    86400 ; Retry
    2419200 ; Expire
    604800) ; Negative Cache TTL
;
@ IN NS ns.grup2.gsx.
@ IN MX 10 correu.grup2.gsx.
ns IN A 172.17.2.1

; correu
correu IN A 172.17.2.3
smtp CNAME correu
pop3 CNAME correu

www.taller IN A 172.17.2.2
www.tenda CNAME www.taller
```

## **INTRANET.db**

```
$TTL 604800
@ IN SOA ns.interna. root.ns.interna. (
    1 ; Serial
    604800 ; Refresh
    86400 ; Retry
    2419200 ; Expire
    604800) ; Negative Cache TTL
;
@ IN NS ns.interna.
ns IN A 192.168.8.1

; pcs
pc1 IN A 192.168.8.2
pc2 IN A 192.168.8.3
pc3 IN A 192.168.8.4
```

## **db.172**

```
$TTL 604800
@ IN SOA ns.grup2.gsx. root.ns.grup2.gsx. (
    1 ; Serial
    604800 ; Refresh
    86400 ; Retry
    2419200 ; Expire
    604800) ; Negative Cache TTL
;
@ IN NS ns.grup2.gsx.
1 IN PTR ns.grup2.gsx.
2 IN PTR www.taller.grup2.gsx.
```

```

2      IN PTR  www.tenda.grup2.gsx.
3      IN PTR  correu.grup2.gsx.
3      IN PTR  smtp.grup2.gsx.
3      IN PTR  pop3.grup2.gsx.

```

### db.192.168.8

```

$TTL 604800
@      IN SOA      ns.interna. root.ns.interna. (
    1      ; Serial
    604800 ; Refresh
    86400  ; Retry
    2419200 ; Expire
    604800 ) ; Negative Cache TTL
;
@      IN NS       ns.interna.
1      IN PTR      ns.interna.
2      IN PTR      pc1.interna.
3      IN PTR      pc2.interna.
4      IN PTR      pc3.interna.

```

### db.192.168.9

```

$TTL 604800
@      IN SOA      ns.interna. root.ns.interna. (
    1      ; Serial
    604800 ; Refresh
    86400  ; Retry
    2419200 ; Expire
    604800 ) ; Negative Cache TTL
;
@      IN NS       ns.interna.
1      IN PTR      ns.interna.
2      IN PTR      pc1.interna.
3      IN PTR      pc2.interna.
4      IN PTR      pc3.interna.

```

A més, afegim en la primer línia del fitxer /etc/reolv.conf, nameserver 127.0.0.1, per poder fer queries des del router a les demés màquines.

### Codi router

```

if [ $# -lt 4 ]
then
    echo "Usem la configuració per defecte"

    #Llistem les interfícies de xarxa
    interfaces=$(ip address show | egrep -e "[0-9]" | cut -d ':' -f2 | tr -d ' ' | sed '/lo/d')
    i1=$(echo $interfaces | cut -d ' ' -f1)
    i2=$(echo $interfaces | cut -d ' ' -f2)

```

```

i3=$(echo $interfaces | cut -d ' ' -f3)

else
    i1=$2
    i2=$3
    i3=$4
fi

config="auto lo\niface lo inet loopback
        \n
        \nauto $i1\niface $i1 inet dhcp
        \n
        \nallow-hotplug $i2\niface $i2 inet static\n\taddress 192.168.16.1\n\tinetmask
255.255.255.0\n\tnetwork 192.168.16.0\n\tbroadcast 192.168.16.255
        \n
        \nallow-hotplug $i3\niface $i3 inet static\n\taddress 192.168.17.1\n\tinetmask
255.255.255.0\n\tnetwork 192.168.17.0\n\tbroadcast 192.168.17.255"

cp -p dhcpd.conf /etc/dhcp/dhcpd.conf

#Afegim les interfícies en /etc/default/isc-dhcp-server
sed -i "s/INTERFACES=\".*\"/INTERFACES=\"$i2, $i3\"/g" "/etc/default/isc-dhcp-server"

#Apaguem les interfícies de xarxa
ifdown $i1 --force
ifdown $i2 --force
ifdown $i3 --force

#escribim canvis en el fitxer /etc/network/interfaces
echo -e $config > /etc/network/interfaces

#Activem el forwarding
echo 1 >/proc/sys/net/ipv4/ip_forward

#escribim les IPs per dur a terme el forwarding de les queries
echo -e "options{\n\tdirectory \"/var/cache/bind\";\n\tforwarders {\" > /etc/bind/named.conf.options
IFS=$'\n'
for line in $(cat /etc/resolv.conf)
do
    if [ "$(echo $line | egrep -e "[0-9].*")" != "" ]
    then
        echo -e "\t$(echo $line | egrep -e "[0-9].*" | cut -d ' ' -f2);\" >>
/etc/bind/named.conf.options
    fi
done
echo -e "\t};\n};\" >> /etc/bind/named.conf.options

cp named.conf.local /etc/bind/named.conf.local
cp DMZ_2.gsx.db /var/cache/bind/DMZ_2.gsx.db
cp INTRANET.db /var/cache/bind/INTRANET.db
cp db.192.168.9 /var/cache/bind/db.192.168.9
cp db.192.168.8 /var/cache/bind/db.192.168.8
cp db.172 /var/cache/bind/db.172

cp db.192.168.250 /var/cache/bind/db.192.168.250
cp VLAN.db /var/cache/bind/VLAN.db

```

```
#reiniciem les interficies de xarxa
ifup $i1
ifup $i2
ifup $i3

#obtenim la ip amb conexió a internet del router
$addres=$(hostname -l | cut -d ' ' -f1)

#permetem el tràfic cap a l'exterior
iptables -I INPUT -j ACCEPT
iptables -t nat -A POSTROUTING -s 192.168.8.0/23 -j SNAT --to $addres
iptables -t nat -A POSTROUTING -s 172.17.2.0/24 -j SNAT --to $addres

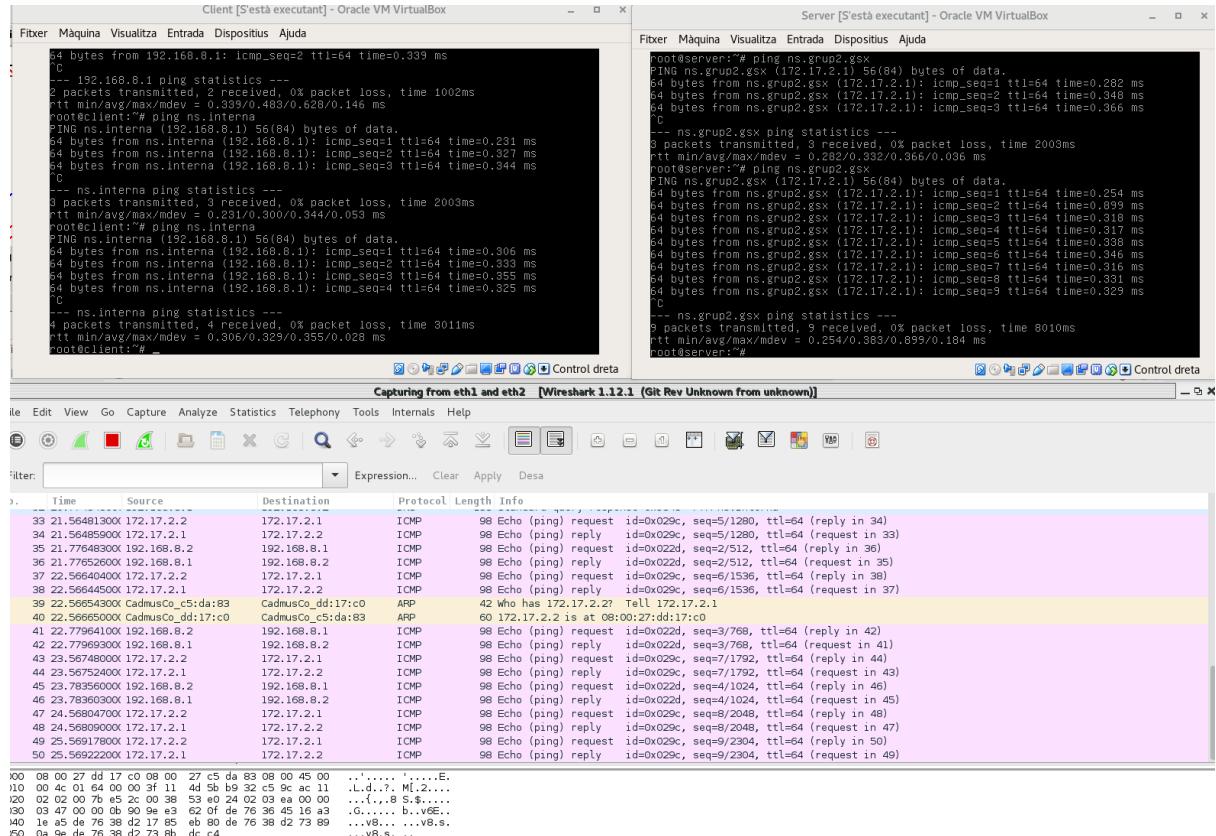
sed -e 'li nameserver 127.0.0.1' /etc/resolv.conf > resolv.conf
cp resolv.conf /etc/resolv.conf
rm resolv.conf

/etc/init.d/bind9 restart
```

En el servidor i client no hi ha cap modificació respecte la fase anterior.

### Joc de proves

En aquesta primera captura, el servidor realitza un ping cap al router mitjançant l'adreça ns.grup2.gsx, de la mateixa forma, el client també fa ping al router mitjançant l'adreça ns.interna. En el router s'observa el trànsit de paquets ICMP deguts als pings del client i servidor.



En aquesta segona, captura s'observa com en fer dig a ns.interna (192.168.8.1) des del client, el servidor DNS usat és el router amb IP 192.168.8.1, per tant, queda demostrat que el servei DNS funciona.

```

root@client:~# dig ns.interna

; <>> DiG 9.10.3-P4-Debian <>> ns.interna
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 37511
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;ns.interna.           IN      A

;; ANSWER SECTION:
ns.interna.        604800  IN      A      192.168.8.1

;; AUTHORITY SECTION:
interna.          604800  IN      NS     ns.interna.

;; Query time: 3 msec
;; SERVER: 192.168.8.1#53(192.168.8.1)
;; WHEN: Mon Apr 09 21:17:12 CEST 2018
;; MSG SIZE  rcvd: 69

```

Proves adicionals que hem realitzat són les de fer ping des del client a [www.taller.grup2.gsx](http://www.taller.grup2.gsx) i [www.tenda.grup2.gsx](http://www.tenda.grup2.gsx) que corresponen a la màquina del servidor amb IP (172.17.2.2).

```

Client [S'està executant] - Oracle VM VirtualBox
Fitxer Mànqua Visualitza Entrada Dispositius Ajuda

;; Query time: 0 msec
;; SERVER: 192.168.8.1#53(192.168.8.1)
;; WHEN: Mon Apr 09 21:23:07 CEST 2018
;; MSG SIZE  rcvd: 98

root@client:~# ping www.taller.grup2.gsx
PING www.taller.grup2.gsx (172.17.2.2) 56(84) bytes of data.
64 bytes from www.taller.grup2.gsx (172.17.2.2): icmp_seq=1 ttl=63 time=0.450 ms
64 bytes from www.taller.grup2.gsx (172.17.2.2): icmp_seq=2 ttl=63 time=0.615 ms
64 bytes from www.taller.grup2.gsx (172.17.2.2): icmp_seq=3 ttl=63 time=0.609 ms
^C
--- www.taller.grup2.gsx ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2012ms
rtt min/avg/max/mdev = 0.450/0.558/0.615/0.076 ms
root@client:~# ping www.tenda.grup2.gsx
PING www.tender.grup2.gsx (172.17.2.2) 56(84) bytes of data.
64 bytes from www.tenda.grup2.gsx (172.17.2.2): icmp_seq=1 ttl=63 time=0.495 ms
64 bytes from www.tenda.grup2.gsx (172.17.2.2): icmp_seq=2 ttl=63 time=0.571 ms
64 bytes from www.tenda.grup2.gsx (172.17.2.2): icmp_seq=3 ttl=63 time=0.580 ms
^C
--- www.tender.grup2.gsx ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2004ms
rtt min/avg/max/mdev = 0.495/0.548/0.580/0.046 ms
root@client:~# 

```

Finalment hem fet ping a google amb l'adreça [www.google.com](http://www.google.com) per demostrar que el servidor DNS implementat permet el forwarding de les queries desconegudes cap al servidor DNS de l'ISP i estàn correctament configurades les iptables, donant servei a internet tant al servidor com al client.

```

root@client:~# ping www.google.com
PING www.google.com (216.58.211.36) 56(84) bytes of data.
64 bytes from www.google.com (216.58.211.36): icmp_seq=1 ttl=61 time=16.3 ms
64 bytes from www.google.com (216.58.211.36): icmp_seq=2 ttl=61 time=19.3 ms
64 bytes from www.google.com (216.58.211.36): icmp_seq=3 ttl=61 time=14.5 ms
64 bytes from www.google.com (216.58.211.36): icmp_seq=4 ttl=61 time=15.6 ms
^C
--- www.google.com ping statistics ---
5 packets transmitted, 4 received, 20% packet loss, time 4021ms
rtt min/avg/max/mdev = 14.556/16.485/19.339/1.777 ms
root@client:~# more /etc/resolv.conf
domain INTRANET
search INTRANET
nameserver 192.168.8.1
root@client:~# _

```

### Part Opcional

Al configurar la VLAN per donar-li també servei DNS a la xarxa el fitxers es modifiquen de la següent forma:

#### **named.conf.local**

```

zone "serveis.interna" {
    type master;
    file "VLAN.db";
};

zone 250.168.192.in-addr.arpa{
    type master;
    file "db.192.168.250";
};

```

#### **VLAN.db**

```

$TTL 604800
@      IN  SOA   ns.serveis.interna. root.ns.serveis.interna. (
                  1          ; Serial
                  604800    ; Refresh
                  86400     ; Retry
                  2419200   ; Expire
                  604800)   ; Negative Cache TTL
;
@      IN  NS    ns.serveis.interna.
@      IN  MX  10 correu.serveis.interna.
ns    IN  A    192.168.150.1

; correu
correu  IN  A    192.168.150.3
smtp    CNAME  correu
pop3    CNAME  correu

www.taller  IN  A    192.168.150.2
www.tenda  CNAME  www.taller

```

## db.192.168.250

```
$TTL 604800
@ IN SOA ns.serveis.interna. root.ns.serveis.interna. (
    1           ; Serial
    604800      ; Refresh
    86400       ; Retry
    2419200     ; Expire
    604800)     ; Negative Cache TTL
;
@ IN NS ns.serveis.interna.
1 IN PTR ns.serveis.interna.
2 IN PTR www.taller.serveis.interna.
2 IN PTR www.tenda.serveis.interna.
3 IN PTR correu.serveis.interna.
3 IN PTR smtp.serveis.interna.
3 IN PTR pop3.serveis.interna.
```